Tarea 3 Redes de Telecomunicaciones

Gonzalo Olvera Monroy

18 de octubre de 2020

1. R4. Describa por qué un desarrollador de aplicaciones podría elegir ejecutar una aplicación sobre UDP en lugar de TCP.

Un desarrollador de aplicaciones puede no querer que su aplicación utilice el control de congestión de TCP, que puede reducir la tasa de envío de la aplicación en momentos de congestión. A menudo, los diseñadores de aplicaciones de telefonía IP y videoconferencia IP eligen ejecutar sus aplicaciones sobre UDP porque quieren evitar el control de congestión de TCP. Además, algunas aplicaciones no necesitan la transferencia de datos confiable proporcionada por TCP.

2. R5. ¿Por qué es que el tráfico de voz y vídeo a menudo se envía a través de TCP en lugar de UDP en la Internet de hoy? (Sugerencia: La respuesta que estamos buscando no tiene nada que ver con el mecanismo de control de congestión de TCP.)

Dado que la mayoría de los firewalls están configurados para bloquear el tráfico UDP, el uso de TCP para el tráfico de video y voz permite que el tráfico atraviese los firewalls.

3. R6. ¿Es posible que una aplicación disfrute de una transferencia de datos fiable incluso cuando la aplicación se ejecuta sobre UDP? Si es así, ¿cómo?

Si. El desarrollador de la aplicación puede transferir datos confiables al protocolo de la capa de la aplicación. Sin embargo, esto requeriría una

cantidad significativa de trabajo y depuración.

4. R12. Visite el applet Go-Back-N Java en el sitio Web adjunto.

a. Haga que la fuente envíe cinco paquetes, y luego detenga la animación antes de que cualquiera de los cinco paquetes llegue al destino. Luego matar el primer paquete y reanudar la animación. Describir lo que sucede.

b. Repita el experimento, pero ahora deje que el primer paquete llegue al destino y elimine el primer reconocimiento. Describa de nuevo lo que sucede.

c. Finalmente, intente enviar seis paquetes. ¿Qué sucede?

Respuestas:

a. La pérdida de paquetes provocó un tiempo de espera después del cual se retransmitieron los cinco paquetes.

b. La pérdida de un ACK no provocó ninguna retransmisión ya que Go-Back-N utiliza agradecimientos acumulativos.

c. El remitente no pudo enviar el sexto paquete porque el tamaño de la ventana de envío está fijado en 5.

5. R13.Repite R12, pero ahora con el applet Java de repetición selectiva. ¿En qué se diferencian la repetición selectiva y el go-back-N?

a. Cuando el paquete se perdió, los cuatro paquetes recibidos fueron almacenados en búfer en el receptor. Después del tiempo de espera, el remitente retransmitió el paquete perdido y el receptor entregó los paquetes almacenados en búfer a la aplicación en el orden correcto.

b. ACK duplicado fue enviado por el receptor para el ACK perdido.

c. El remitente no pudo enviar el sexto paquete ya que el tamaño de la ventana de envío se fija en 5. Cuando se perdió un paquete, GO-Back-N retransmitió todos los paquetes mientras que Selective Repeat retransmitió el paquete perdido solamente. En caso de acuse de recibo perdido, la repetición selectiva envió un duplicado ACK y como GO-Back-N utilizó el acuse de recibo acumulativo, de modo que el duplicado ACK era innecesario.

6. R14.¿Verdadero o falso?

a. Host A está enviando a Host B un archivo grande sobre una conexión TCP. Asuma que el Host B no tiene datos para enviar al Host A. El Host B no enviará acuses de recibo al Host A porque el Host B no puede encasillar los acuses de recibo de los datos.

b. El tamaño del rwnd TCP nunca cambia a lo largo de la duración de la conexión.

c. Supongamos que Host A está enviando a Host B un archivo grande sobre una conexión TCP. El número de bytes no reconocidos que A envía no puede exceder el tamaño del buffer de recepción.

d. Supongamos que el Host A está enviando un archivo grande al Host B sobre una conexión TCP. Si el número de secuencia para un segmento de esta conexión es m, entonces el número de secuencia para el segmento posterior será necesariamente $m + 1$.

e. El segmento TCP tiene un campo en su encabezado para rwnd.

f. Supongamos que el último SampleRTT en una conexión TCP es igual a 1 segundo. El valor actual de TimeoutInterval para la conexión será necesariamente ≥ 1 segundo.

g. Supongamos que Host A envía un segmento con número de secuencia 38 y 4 bytes de datos sobre una conexión TCP al Host B. En este mismo segmento el número de reconocimiento es necesariamente 42.

Respuesta:

a. Falso

b. Falso

c. Verdadero

d. Falso

e. Verdadero

f. Falso

g. Falso

7. R15. Supongamos que Host A envía dos segmentos TCP de vuelta a Host B sobre una conexión TCP. El primer segmento tiene el número de secuencia 90; el segundo tiene el número de secuencia 110.

a. ¿Cuántos datos hay en el primer segmento?

b. Supongamos que el primer segmento se pierde pero el segundo segmento llega a B. En el reconocimiento que el Host B envía al Host A, ¿cuál será el número de reconocimiento?

Respuesta:

a. 20 bytes

b. ack number = 90

8. R17. Supongamos que dos conexiones TCP están presentes sobre algún enlace de cuello de botella de tasa R bps. Ambas conexiones tienen un archivo enorme para enviar (en la misma dirección sobre el enlace de cuello de botella). Las transmisiones de los archivos comienzan al mismo tiempo. ¿Qué velocidad de transmisión le gustaría dar a TCP a cada una de las conexiones?

R/2

9. R18. ¿Verdadero o falso? Considere el control de congestión en TCP. Cuando el temporizador expira en el remitente, el valor de ssthresh se establece a la mitad de su valor anterior.

Falso, se establece a la mitad del valor actual de la ventana de congestión.

10. P22. Considere el protocolo GBN con un tamaño de ventana del remitente de 4 y un rango de números de secuencia de 1.024. Supongamos que en el momento t, el siguiente paquete en orden que el receptor está esperando tiene un número de secuencia de k. Supongamos que el medio no reordena los mensajes. Responda a las siguientes preguntas:

a. ¿Cuáles son los posibles conjuntos de números de secuencia dentro de la ventana del remitente en el tiempo t? Justifique su respuesta.

b. ¿CuÃ¡les son todos los valores posibles del campo ACK en todos los mensajes posibles que se están propagando al remitente en el momento t? Justifique su respuesta.

Respuesta:

a. Aquí tenemos un tamaño de ventana de N = 3. Suponga que el receptor ha recibido el paquete k - 1 y ha ACKED ese y todos los demás paquetes anteriores. Si el remitente ha recibido todos estos ACK, la ventana del remitente es [k, k + N - 1]. Supongamos a continuación que el remitente no ha recibido ninguno de los ACK. En este segundo caso, la ventana del remitente contiene k - 1 y los N paquetes hasta k - 1 inclusive. La ventana del remitente es entonces [k - N, k - 1]. Según estos argumentos, la ventana de remitentes es de tamaño 3 y comienza en algún lugar del rango [k - N, k].

b. Si el receptor está esperando el paquete k, entonces ha recibido (y ACKed) el paquete k - 1 y los paquetes N - 1 antes de eso. Si ninguno de esos N ACKs ha sido aún recibido por el remitente, entonces los mensajes ACK con valores de [k - N,k - 1] todavía pueden estar propagándose.Debido a que el remitente ha enviado paquetes [k - N, k - 1], debe ser el caso de que el remitente ya haya recibido un ACK por k - N - 1. Una vez que el receptor ha enviado un ACK para k - N - 1 nunca enviará un ACK que es menos que k - N - 1. Así, el rango de valores de ACK en - vuelo puede variar de k - N - 1 a k - 1.

11. P23. Considere los protocolos GBN y SR. Suponga que el espacio del número de secuencia es de tamaño k. ¿Cuál es la ventana de remitente más grande permitida que evitará la aparición de problemas como que en la Figura 3.27 para cada uno de estos protocolos?

Con el fin de evitar el escenario de la Figura 3.27, queremos evitar que el borde delantero de la ventana del receptor (es decir, el que tiene el número de secuencia "más alto") se envuelva en el espacio de número de secuencia y se superponga con el borde posterior (el que tiene el número de secuencia "más bajo.[en] la ventana del remitente). Es decir, el espacio del número de secuencia debe ser lo suficientemente grande como para caber toda la ventana del receptor y toda la ventana del remitente sin esta condición de superposición. Así que - necesitamos determinar qué tan grande es el rango de números de secuencia que pueden ser cubiertos en un momento dado por las ventanas del receptor y del remitente.
Supongamos que el número de secuencia más bajo - que el receptor está esperando es el paquete m. En este caso, su ventana es [m,m+w - 1] y ha recibido (y ACKed) el paquete m - 1 y el w - 1 paquetes antes de eso, donde w es el tamaño de la ventana. Si ninguno de esos w ACKs han sido aún recibidos por el remitente, entonces los mensajes ACK con valores de [m - w,m - 1] todavía pueden estar propagándose. Si el remitente no ha recibido ACKs con estos números ACK, la ventana del remitente sería [m - w, m - 1].
Por lo tanto, el borde inferior de la ventana del remitente es m - w, y el borde delantero de la ventana del receptor es m+w - 1. Para que el borde delantero de la ventana del receptor no se superponga con el borde posterior de la ventana del remitente, el espacio de número de secuencia debe ser lo suficientemente grande para acomodar números de secuencia de 2w. Es decir, el espacio del número de secuencia debe ser al menos el doble del tamaño de la ventana, k $\geq$2w.

12. P25. Hemos dicho que una aplicación puede elegir UDP para un protocolo de transporte porque UDP ofrece un control de aplicación más fino (que TCP) de qué datos se envían en un segmento y cuándo.

a. ¿Por qué una aplicación tiene más control sobre los datos que se envían en un segmento?

b. ¿Por qué una aplicación tiene más control sobre cuándo se envía el segmento?

Respuesta:

a. Considere enviar un mensaje de aplicación sobre un protocolo de transporte. Con TCP, la aplicación escribe datos al búfer de envío de conexión y TCP tomará bytes sin necesidad de poner un solo mensaje en el segmento TCP; TCP puede poner más o menos un solo mensaje en un segmento. UDP, por otro lado, encapsula en un segmento lo que la aplicación le da; de modo que, si la aplicación le da a UDP una era de desorden de aplicaciones, este mensaje será la carga útil del segmento UDP. Así, con UDP, una aplicación tiene más control de qué datos se envían en un segmento.

b. Con TCP, debido al control de flujo y el control de congestión, puede haber un retraso significativo desde el momento en que una aplicación escribe datos a su buffer de envío hasta cuando los datos se dan a la capa de red. UDP no tiene retrasos debido al control de flujo y control de congestión.

13. P27. El Host A y B se comunican a través de una conexión TCP, y el Host B ya ha recibido de A todos los bytes a través del byte 126. Supongamos que el Host A envía dos segmentos al Host B espalda con espalda. El primer y segundo segmentos contienen 80 y 40 bytes de datos, respectivamente. En el primer segmento, el número de secuencia es 127, el número de puerto de origen es 302, y el número de puerto de destino es 80. Host B envía un acuse de recibo cada vez que recibe un segmento de Host A.

a. En el segundo segmento enviado desde el Host A a B, ¿cuál es el número de secuencia, el número de puerto de origen y el número de puerto de destino?

b. Si el primer segmento llega antes del segundo segmento, en el reconocimiento del primer segmento de llegada, ¿cuál es el número de reconocimiento, el número de puerto de origen y el número de puerto de destino?

c. Si el segundo segmento llega antes del primer segmento, en el reconocimiento del primer segmento que llega, ¿cuál es el número de reconocimiento?

d. Supongamos que los dos segmentos enviados por A llegan en orden a B. El primer reconocimiento se pierde y el segundo reconocimiento llega después del primer intervalo de tiempo. Dibuje un diagrama de tiempo, mostrando estos segmentos y todos los demás segmentos y agradecimientos enviados. (Asuma que no hay pérdida de paquetes adicional.) Para cada segmento en su figura, proporcione el número de secuencia y el número de bytes de datos; para cada reconocimiento que agregue, proporcione el número de reconocimiento.

Respuesta:

a. En el segundo segmento de Host A a B, el número de secuencia es 207, el número de puerto de origen es 302 y el número de puerto de destino es 80.

b. Si el primer segmento llega antes del segundo, en el acuse de recibo del primer segmento de llegada, el número de acuse de recibo es 207, el número de puerto de origen es 80 y el número de puerto de destino es 302.

c. Si el segundo segmento llega antes del primer segmento, en el reconocimiento del primer segmento que llega, el número de acuse de recibo es 127, indicando que todavía está esperando los bytes 127 y siguientes.

d.

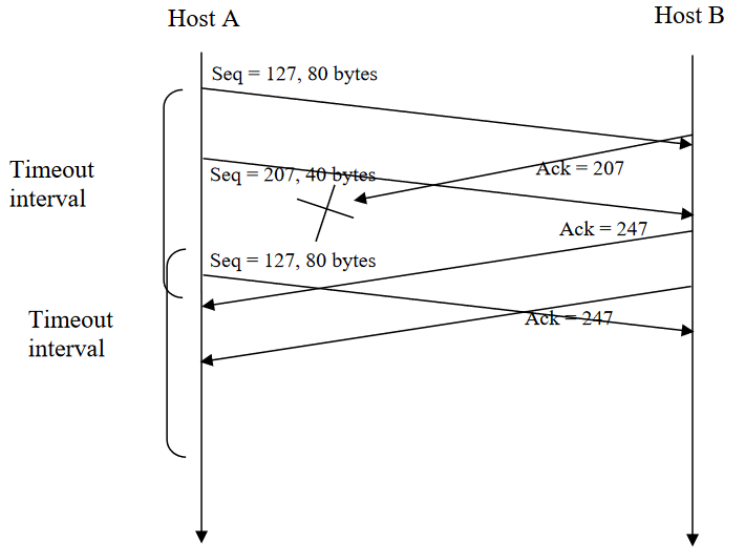

Figura 1:

14. P28. El host A y B están conectados directamente con un enlace de 100 Mbps. Hay una conexión TCP entre los dos hosts, y Host A está enviando a Host B un archivo enorme sobre esta conexión. Host A puede enviar sus datos de la aplicación a su zócalo TCP a una velocidad de hasta 120 Mbps, pero Host B puede leer de su búfer de recepción TCP a una velocidad máxima de 50 Mbps. Describir el efecto del control de flujo TCP.

Dado que la capacidad del enlace es de solo 100 Mbps, la velocidad de envío del host A puede ser como máximo de 100 Mbps. Aún así, el host A envía datos al búfer de recepción mÃ¡s rápido de lo que el host B puede eliminar datos del búfer. El búfer de recepción se llena a una velocidad de aproximadamente 40 Mbps. Cuando el búfer está lleno, el Host B le indica al Host A que deje de enviar datos configurando RcvWindow = 0.
Host A entonces deja de enviar hasta que recibe un segmento TCP con RcvWindow > 0. Por lo tanto, el host A detendrá repetidamente un envío d start en función de los valores RcvWindow que recibe del Host B. En promedio, la tasa a largo plazo a la que el Host A envía datos al Host B

como parte de esta conexión no es más de 60 Mbps.

15. P29. Las cookies SYN se discutieron en la sección 3.5.6 .

a. ¿Por qué es necesario que el servidor utilice un número de secuencia inicial especial en la SYNACK?

b. Supongamos que un atacante sabe que un host de destino utiliza cookies SYN. ¿Puede el atacante crear conexiones semiabiertas o totalmente abiertas simplemente enviando un paquete ACK al destino? ¿Por qué o por qué no?

c. Supongamos que un atacante recoge una gran cantidad de números de secuencia iniciales enviados por el servidor. ¿Puede el atacante hacer que el servidor cree muchas conexiones completamente abiertas enviando ACKs con esos números de secuencia iniciales? ¿Por qué?

Respuesta:

a. El servidor utiliza un número de secuencia inicial especial (que se obtiene a partir del hash de las IPs y puertos de origen y destino) para defenderse del ataque SYN FLOOD.

b. No, el atacante no puede crear conexiones semiabiertas o totalmente abiertas simplemente enviando paquetes ACK al destino. Las conexiones semiabiertas no son posibles ya que un servidor que utiliza cookies SYN no mantiene variables de conexión y buffers para ninguna conexión antes de que se establezcan conexiones completas. Para establecer conexiones completamente abiertas, un atacante debe conocer el número de secuencia inicial especial correspondiente a la dirección IP de origen (falsificada) del atacante. Este número de secuencia requiere el número "secreto"que cada servidor utiliza. Dado que el atacante no conoce este número secreto, no puede adivinar el número de la secuencia inicial.

c. No, el sever simplemente puede añadir una marca de tiempo al calcular esos números de secuencia iniciales y elegir un valor de tiempo para vivir para esos números de secuencia, y descartar los números de secuencia iniciales expirados incluso si el atacante los repite.

16. P33. En la Sección 3.5.3, discutimos la estimación de RTT de TCP. ¿Por qué crees que TCP evita medir el SampleRTT para segmentos retransmitidos?

Veamos lo que podría estar mal si TCP mide SampleRTT para un segmento retransmitido. Supongamos que la fuente envía el paquete P1, el temporizador para P1 expira, y la fuente entonces envía P2, una nueva copia del mismo paquete. Supongamos además que la fuente mide SampleRTT para P2 (el paquete retransmitido). Finalmente supongamos que poco después de transmitir P2 llega un acuse de recibo para P1. La fuente tomará erróneamente este reconocimiento como un reconocimiento para P2 y calculará un valor incorrecto de SampleRTT.
Veamos lo que podría estar mal si TCP mide SampleRTT para un segmento retransmitido. Supongamos que la fuente envía el paquete P1, el temporizador para P1 expira, y la fuente entonces envía P2, una nueva copia del mismo paquete. Supongamos además que la fuente mide SampleRTT para P2 (el paquete retransmitido). Finalmente supongamos que poco después de transmitir P2 llega un acuse de recibo para P1. La fuente tomará erróneamente este reconocimiento como un reconocimiento para P2 y calculará un valor incorrecto de SampleRTT .

17. P40. Considere la Figura 3.58 . Asumiendo que TCP Reno es el protocolo que experimenta el comportamiento mostrado arriba, responda las siguientes preguntas. En todos los casos, debe proporcionar una breve discusión que justifique su respuesta.

a. Identifique los intervalos de tiempo en los que el inicio lento de TCP está funcionando.

b. Identifique los intervalos de tiempo en los que funciona la prevención de congestión de TCP.

c. Después de la 16a ronda de transmisión, ¿se detecta la pérdida de segmento mediante un ACK triple duplicado o por un tiempo de espera?

d. Después de la 22 ronda de transmisión, ¿se detecta la pérdida

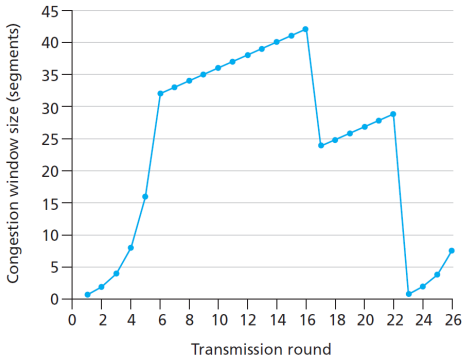

Figura 2: TCP window size as a function of time

de segmento por un triple duplicado ACK o por un tiempo de espera?

e. ¿Cuál es el valor inicial de ssthresh en la primera ronda de transmisión?

f. ¿Cuál es el valor de ssthresh en la 18 ronda de transmisión?

g. ¿Cuál es el valor de ssthresh en la 24 ronda de transmisión?

h. ¿Durante qué ronda de transmisión se envÃa el segmento 70?

i. Suponiendo que se detecta la pérdida de un paquete después de la $26a$ ronda mediante la recepción de un triple ACK duplicado, ¿cuáles serán los valores del tamaño de la ventana de congestión y de ssthresh?

j. Suponga que se usa TCP Tahoe (en lugar de TCP Reno), y asuma que los ACK duplicados triples se reciben en la 16 ronda. ¿Cuáles son los ssthresh y el tamaño de la ventana de congestión en la 19 ronda?

k. De nuevo supongamos que se utiliza TCP Tahoe, y hay un evento de tiempo de espera en la ronda 22. ¿Cuántos paquetes se han enviado desde la ronda 17 hasta la ronda 22?

Respuesta:

a. TCP slowstart está operando en los intervalos [1,6] y [23,26]

b. TCP congestion avoidance está operando en los intervalos [6,16] y [17,22]

c. Después del 16 ronda de transmisión, la pérdida de paquetes se reconoce mediante un triple ACK duplicado. Si hubiera un tiempo de espera, el tamaño de la ventana de congestión se habría reducido a 1.

d. Después de la 22 ronda de transmisión, la pérdida de segmento se detecta debido al tiempo de espera, y por lo tanto el tamaño de la ventana de congestión se establece en 1.

e. El umbral es inicialmente 32, ya que es en este tamaño de ventana donde se detiene el slow start y comienza la evitación de la congestión.

f. El umbral se establece en la mitad del valor de la ventana de congestión cuando se detecta la pérdida de paquetes. Cuando se detecta una pérdida durante la ronda 16 de transmisión, el tamaño de las ventanas de congestión es 42. Por tanto, el umbral es 21 durante la ronda 18 de transmisión.

g. El umbral se establece a la mitad del valor de la ventana de congestión cuando se detecta la pérdida de paquetes. Cuando la pérdida se detecta durante la ronda de transmisión 22, el tamaño de las ventanas de congestión es 29. Por lo tanto el umbral es 14 (tomando el piso inferior de 14.5) durante la 24 ronda de transmisión.

h. Durante la primera ronda de transmisión, se envía el paquete 1; el paquete 2-3 se envía en la segunda ronda de transmisión; los paquetes 4-7 se envían en la tercera ronda de transmisión; los paquetes 8-15 se envían en la cuarta ronda de transmisión; los paquetes 16-31 se envían en la quinta ronda de transmisión; los paquetes 32-63 se envían en la 6 ronda de transmisión; los paquetes 64-96 se envían en la 7 ronda de transmisión. Así, el paquete 70 se envía en la séptima ronda de transmisión.

i. El umbral se fijará a la mitad del valor actual de la ventana de congestión (8) cuando se produzca la pérdida y la ventana de congestión se fijará al nuevo valor de umbral + 3 SMS . Así, los nuevos valores del umbral y de la ventana serán 4 y 7 respectivamente.

j. El umbral es 21 y el tamaño de la ventana de congestión es 1.

k. Ronda 17, 1 paquete;
Ronda 18, 2 paquetes;
Ronda 19, 4 paquetes;
Ronda 20, 8 paquetes;
Ronda 21, 16 paquetes;
Ronda 22, 21 paquetes.
Entonces, el número total es 52.